NUWC-NPT Technical Report 11,050
30 November 1998

An Adaptive Mesh Technique for Finding the Bistatic Specular Point on a Convex Hull

Thomas A. Wettergren
Torpedo Systems Technology Department

Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

This report was prepared under contract number N00024-98-C-6107, "Common Broadband Advanced Sonar System (CBASS) Program," principal investigator Denise P. Campbell (Code 8123). The sponsoring activity is the Program Executive Officer for Undersea Warfare (PMS-404), program manager M. Jacobs.

The technical reviewer for this report was Charles E. Rosenthal (Code 8214).

Reviewed and Approved: 30 November 1998

James C. S. Meng
Head, Torpedo Systems Technology Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 November 1998	Final

4. TITLE AND SUBTITLE

An Adaptive Mesh Technique for Finding the Bistatic Specular Point on a Convex Hull

5. FUNDING NUMBERS

PR B34820

6. AUTHOR(S)

Thomas A. Wettergren

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center Division
1176 Howell Street
Newport, RI 02841-1708

8. PERFORMING ORGANIZATION REPORT NUMBER

TR 11,050

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Program Executive Officer
Undersea Warfare (PMS-404)
2531 Jefferson Davis Highway
Arlington, VA 22242-5169

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A new adaptive mesh technique for searching for a bistatic specular point on a convex hull surface has been developed for the High Frequency Target Acoustic Model (HFTAM). The existing method of bistatic specular point searching in the HFTAM uses a bisection search with numerical evaluations of a transcendental equation at each bisection step. This method has been shown to lack computational robustness. An adaptive mesh technique has been implemented with the HFTAM and has been shown to be more robust than the bisection technique.

This report describes the new version of the HFTAM and the equations used in the model; it also provides guidelines for software implementation.

14. SUBJECT TERMS

Acoustic Models
Signal Processing
Underwater Acoustics

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

AN ADAPTIVE MESH TECHNIQUE FOR FINDING THE BISTATIC SPECULAR POINT ON A CONVEX HULL

INTRODUCTION

One of the most computationally expensive routines in the Naval Undersea Warfare Center's (NUWC's) High Frequency Target Acoustic Model (HFTAM) is the routine that searches for a bistatic specular point along a hull-like surface. This routine is the first part of the specular scattering calculation, and it computationally determines the point along the hull surface that minimizes the distance from the source to the surface point to the receiver. This point will be unique for a convex shape and will be the point whose normal bisects the vectors from the point to the source and from the point to the receiver. This bistatic specular point is used by other routines to compute the specular scattering from the hull-like structure. Monostatic geometries are not handled separately and are only viewed as a case where the source and receiver happen to be collocated. The computational routine that finds the specular point treats the hull-like geometry description with the source and receiver locations as input and returns the coordinates of the specular scattering point as output. For the remainder of this report, the process of finding this specular point for a fixed source/receiver pair will be called a hull search.

This report begins with a description of the equations used to model convex hulls, as they are an integral component of the solution techniques. Then, a detailed explanation of the current bisection hull search technique and its performance is presented. This is followed by a description of the new adaptive mesh hull search technique and its performance relative to the existing technique. Finally, a software implementation guide is presented for incorporating the new search routine into the HFTAM.

GEOMETRY OF THE CONVEX HULLS

The surfaces represented as convex hulls are many structures that can be internal or external to a target, have complex curvature, and are not necessarily represented by canonical shapes. The only restriction for fitting a geometrical shape to these surfaces is that the surface must be representable by a convex shape (or a portion thereof). Since the majority of the shapes of interest are submarine-like (i.e., shaped like a perturbed cylindrical shell), it is advantageous to represent the geometry by an equation that is implicit in the coordinates and is of the form

$$f(y, m_1(x), m_2(x), \ldots) + g(z, n_1(x), n_2(x), \ldots) = 0. \qquad (1)$$

In this form, the functions m_j and n_j represent parameters that vary along the length of the hull (hence their explicit dependence on x). By using a representation of this form, it is a direct procedure to find parameters that fit a given geometry to the equations by fitting "slices" of the y-z plane at fixed values of x (i.e., identify values of $m_j(x_1)$, $m_j(x_2)$, etc., for a large set of x locations), and then allowing the m_j and n_j parameters to vary smoothly as a function of x. In this context, "smoothly" implies continuity through at least the second derivatives (C^2 continuity).

To represent the types of curvature often seen in these y-z slices, the HFTAM uses a geometric expression for each slice that is represented by a hyperellipse:[1]

$$\left(\frac{y}{b(x)}\right)^{p(x)} + \left(\frac{z-d(x)}{a(x)}\right)^{p(x)} = 1, \tag{2}$$

where the parameter d allows a non-zero offset of the slice in the z-direction. An example of a surface defined in this way is shown in figure 1. The expressions for the parameters defining this particular surface can be found in appendix A.

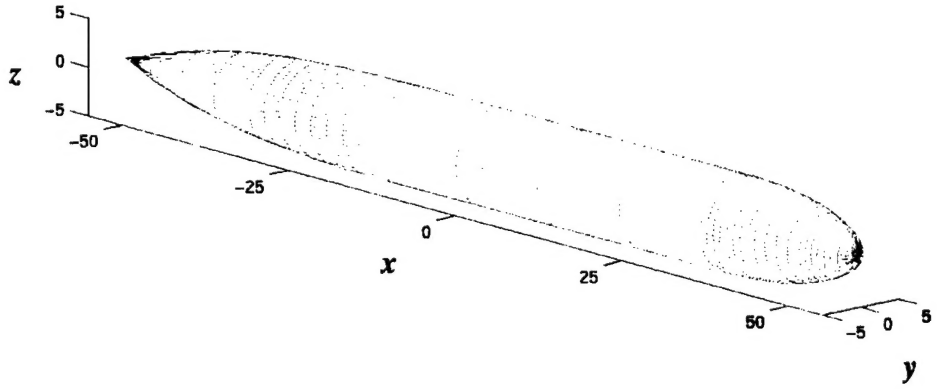

Figure 1. Sample Convex Hull Represented by the Hyperelliptic Formulation

To allow for a greater variety of surfaces, the hyperellipse expression can be supplemented by a set of cutouts. These are regions that are defined in x-z space and are flagged as existing on the mathematical convex hull but do not exist on the real surface. In this way, partial convex hull pieces may be used to represent complex geometries. Hence, the specular point is computed as if it were a complete convex hull, and then computational logic determines whether or not a real specular point exists (which occurs when the computed point is not in the flagged region). An example of a surface defined with a cutout is shown in figure 2. In this figure, the red region on the bottom and the blue region on the top are separated by a cutout shown by a gray slice.

This example shows how a piece of a very complex curved surface can be represented by a convex hull with associated cutout regions. In this case, the red section could be a deck structure that is defined separately from a main hull, thus allowing a

nonconvex geometry to be represented as the union of multiple pieces of convex geometries. Also, the hyperelliptic exponent $p > 2$ for the red section (becoming as large as 20), giving the convex hull section a squared-off deck-like appearance. In this way, virtually any geometry that is seen in an undersea structure can be represented by a series of (potentially partial) convex hull sections. The expressions for the parameters that define the surface and the cutout region shown in figure 2 are found in appendix B.

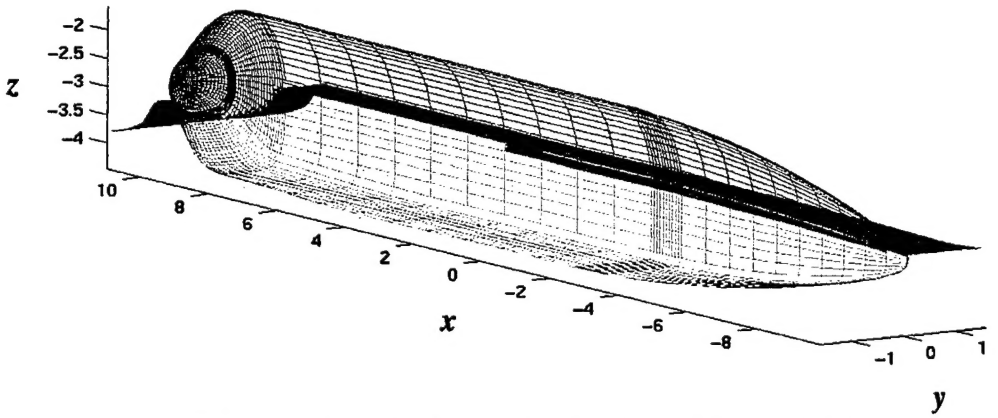

Figure 2. Sample Convex Hull with a Cutout Region

BISECTION SEARCH APPROACH

The current technique that is implemented in HFTAM for finding the specular point was provided by Lengua[2,3] and is briefly reviewed here. In this algorithm, a one-dimensional bisection search[4] is performed in the x-direction. At each step of the bisection search, the source and receiver points are projected onto the y-z plane. An analytical expression for the minimization of the distance from the source to hull to receiver within this plane is then evaluated.

The analytical minimization function is obtained as follows. For a candidate x location given by $x = x_c$, the total distance from the source to a point on the hull is given by

$$\rho_s(y,z) = \sqrt{(z_s - z)^2 + (y_s - y)^2}, \tag{3}$$

where the subscript s corresponds to the source's coordinates. The distance to the receiver is given by a similar expression for ρ_r. The total distance is minimized for a given x location by first using equation (2) for the fixed x value to solve for y in terms of z as

$$y = b(x_c)\left[1 - \left(\frac{z - d(x_c)}{a(x_c)}\right)^{p(x_c)}\right]^{1/p(x_c)}. \tag{4}$$

With this definition for y, the minimization problem is written in terms of the single unknown z by first setting the distance variation to zero for a minimum as

$$\frac{d}{dz}(\rho_s + \rho_r) = 0. \tag{5}$$

The minimization expression in equation (5) is solved by combining with equation (3) to get

$$0 = \left(\frac{z_s - z}{\rho_s}\right) + \left(\frac{z_r - z}{\rho_r}\right) + \left(\frac{dy}{dz}\right)\left[\left(\frac{y_s - y}{\rho_s}\right) + \left(\frac{y_r - y}{\rho_r}\right)\right]. \tag{6}$$

Since y is a function of z in equation (4), this expression has a single unknown quantity z. However, the expression for z in equation (6) will be transcendental rather than a simple polynomial (due to both $y(z)$ and the derivative of y with respect to z) and has to be solved by a numerical procedure. The HFTAM program currently uses a bisection search for this transcendental evaluation.

This technique is mathematically elegant but computationally costly. Furthermore, there is no guarantee of convergence of the algorithm, because the convergence is dependent on the numerical simplicity of the analytical minimization problem. If the points found at each x_c vary in a simple manner as the algorithm moves through x values, there is good convergence. However, if the points found at each x_c vary in a complicated way (even if this is only a numerical phenomenon), the convergence may be poor or even nonexistent.

The bisection search technique performs well when the hull shape is nearly elliptical (i.e., $p \cong 2$) since these cases tend to have a simple analytical expression for the minimum distance point for a given slice. However, for the cases where the convex hull geometry is used to represent complex shapes other than hulls, and also for the hulls of many diesel-electric submarines, the hull shapes are more squared-off at the corners ($p >> 2$) rather than elliptical. To create an accurate representation of these nonelliptic geometries, the hyperellipse formulation uses an arbitrary exponential $p(x)$, which may vary along the length of the hull. It is in these cases that the bisection search algorithm has difficulty in converging and often does not converge in the maximum number of iterations or fails to converge due to the complexity of the changes in the parameter p over the search length. Furthermore, these cases tend to be extremely computationally intense even when they do converge to a solution. This computational complexity is caused by the need to perform a secondary bisection search at each candidate x value as well as performing a large number of noninteger power evaluations in the analysis of equation (4) and its derivative. Thus, there are costly computations that occur in both the evaluation of a given test point and the iteration between test points. It should be noted,

however, that the computational complexity is greatly reduced when $p = 2$. In that case, the expression in equation (6) can be rewritten as a very large polynomial equation in z, which can be solved quickly and accurately.

To illustrate the computational complexity of these routines when the hull shape is nonelliptical (i.e., $p \neq 2$), the hull search evaluations that are performed for two different submarine-like models must be considered. The first case uses a model called *TestSub*, which contains both hull and internal hull-like structures, all of which have p identically equal to two. The second case is with a model called *Omega*, which uses p values that not only are different from two but also vary along the length of the hull. Furthermore, this model has internal hull-like structures that have p values other than two. To perform an accurate measure of the computational complexity of these two models, they were both run on the same computer (a 130-MHz IBM RS/6000) over the aspect range of –180° to +180° at a 0.00° elevation angle in a monostatic geometry. The number of hull search calls varies as a function of aspect as various internal structures become hidden from the acoustic path. Thus, the total time spent in the hull search routine over all aspect runs was tallied and divided by the total number of hull search calls to give a value for the average time per hull search. The results are shown in table 1. It is clear from table 1 that the hull search routine with elliptical curves is much faster than the routine for nonelliptical curves. In fact, the nonelliptical routine had an average search time that was over 12 times longer than that for the elliptical hull. Similar results have been observed for other models. Thus, the bisection hull search routine has been shown to be very slow for nonelliptical hulls.

Table 1. Performance of the Bisection Hull Search

Model	Number of Hull Searches	Total Time in Hull Searches	Average Time per Hull Search
TestSub	334	0.20 sec	0.60 msec
Omega	543	4.15 sec	7.64 msec

For real-time (and even for fast wall-time) simulation purposes, it is very important to make all of the routines in a model as fast as possible while maintaining a high level of fidelity. Furthermore, a guarantee of convergence at all aspects (where specular points exist) is a requirement for simulations because the model is used only for one specific aspect at a time. In the context of this report, this is referred to as the robustness of the numerical algorithm. The robustness of the bisection search algorithm was addressed by examining two runs of the Omega model, i.e., one at 0.00° elevation angle and a second at 0.01°. The results of these two runs for the outer hull target strength are shown in figures 3 and 4. It is obvious from these plots that the 0.00° elevation run is missing the beam aspect behavior that is shown in the 0.01° elevation run. Some analysis of the runs found this to be caused by a lack of convergence of the hull search routine in the former run, which caused the target strength to be nonexistent. This extreme sensitivity to a minor change in elevation angle is not a physical effect but merely a numerical artifact that demonstrates the lack of robustness of the bisection hull search technique.

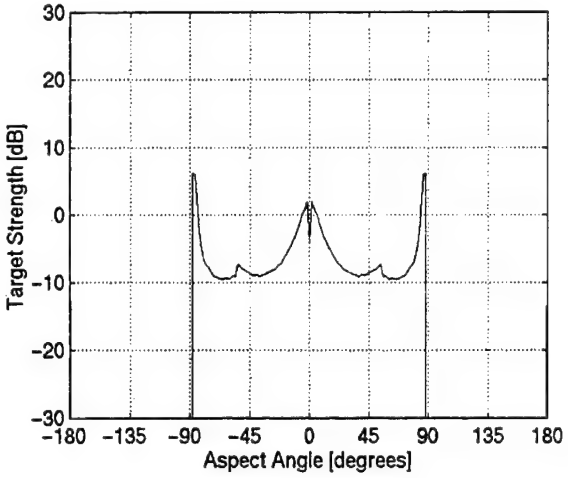

Figure 3. Target Strength for Omega Model Hull with Bisection Hull Search at 0.00° Elevation Angle

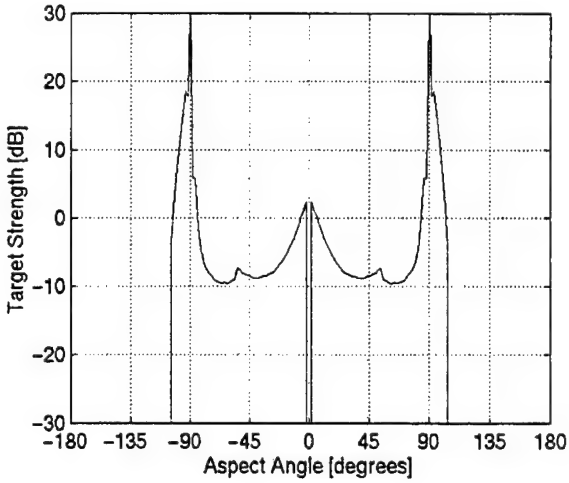

Figure 4. Target Strength for Omega Model Hull with Bisection Hull Search at 0.01° Elevation Angle

The robustness issue is a primary concern only in simulation applications; it is much less important when plots of behavior over a range of aspects is the required output, such as for target strength plots or "hourglass" echo plots. However, robustness is a critical issue when the output of the model is going to be used as input for a processing scheme, as is the case in most simulation applications. In such cases, individual runs (specific aspect angles) are performed separately and the individual run results are more important than the composite. A numerical search technique that is based on algebraic expressions with less complexity may improve this robustness issue. When such an algorithm is developed with equivalent or better speed performance than the bisection method, it will be a practical improvement.

GEOMETRIC MESH SEARCH APPROACH

A new algorithm for the bistatic hull search problem has been developed. This algorithm was created to correct the robustness and speed problems of the bisection search algorithm for simulation purposes. Robustness issues arise when the geometry along the hull shape either changes extremely rapidly (sharp corners) or very slowly (cylindrical sections) and the bisection algorithm has difficulty converging. Furthermore, the speed of a hull search routine used in simulations needs to be relatively insensitive to the geometric complexity of the specific structure. This type of robustness is a requirement of the user community to ensure that the computational model's speed is stable, regardless of the input parameters of a particular run.

To handle these robustness issues, a search method was developed based on local adaptive refinement of geometric meshes along the convex hull's surface. The advantage of a geometric mesh is that the details of a surface's complexity lie in the definition of the

mesh points, and the surface between the mesh points is represented by simple curves. Thus, once the mesh is generated, all of the complicated equations defining the surface have been replaced by a table of mesh points.

For the acoustic scattering problem, the mesh is defined to a level of detail such that the local radii of curvature at a specular point are determined to an accuracy of at least a wavelength squared (λ^2). This requirement is driven by the specular scattering equation from the Geometric Theory of Diffraction[5] and its numerical evaluation. This level of detail can be guaranteed only if the space between nearest mesh points is no greater than the wavelength λ. Such a mesh can be prohibitively large; for example, consider a cylinder with a 5-m radius and a length of 50 m being ensonified at 15 kHz. The minimal number of mesh points to guarantee a one-wavelength maximum distance between nearest points is 157,394 (this does not even account for the endcaps). Such a large number of mesh points is not practical to generate; therefore, a meshing scheme based on local mesh refinement has been developed.

Adaptive mesh refinement is practical for a specular point searching routine because the specular points are a local phenomena, and the mesh only needs to be refined in a neighborhood of the specular point. To locally determine the specular point at a given level of refinement, the mesh is evaluated point-wise and the minimum combined distance from source to mesh point to receiver is tabulated. The point with the smallest combined distance is the closest point to specular in the given level of refinement for the mesh points evaluated. To further refine the mesh, the new search area around the chosen point must be made large enough to guarantee convergence as the mesh refines. In a worst case, the mesh can be refined by at least a level of one-fourth (in area) at each step. In this scenario, the specular point will be known to within an accuracy of

$$\mathrm{dist}\left(SP - SP^*\right) \le \sqrt{\left(1/4\right)^n A_0} \tag{7}$$

after n steps of refining, where A_0 is the initial convex hull surface area, SP is the exact specular point, SP^* is the computed specular point, and $dist(\,)$ is the standard L_2 distance norm.

The locally adaptive meshing technique that has been developed for the hull search problem uses a mapping of the convex hull surface into a two-dimensional space. The mapping was specifically derived for hyperelliptic convex hulls represented by equation (2). The mapping that is used in this algorithm translates between the physical (x,y,z) space and the (x,θ) space, where x is the same in both domains. This mapping is given by

$$y(x,\theta) = \pm b(x)\cdot\left(\frac{|\tan\theta|}{\beta(\theta)}\right), \quad z(x,\theta) = d(x) \pm a(x)\cdot\left(\frac{1}{\beta(\theta)}\right), \tag{8}$$

where

$$\beta(\theta) = \left[1 + |\tan\theta|^{p(x)}\right]^{1/p(x)}, \tag{9}$$

and the inverse mapping is given by

$$\theta = \arctan\left(\frac{y - a(x)}{[z - d(x)]\cdot b(x)}\right). \tag{10}$$

In equation (8), the sign in the expression for y is chosen to be the same as the sign of θ, and the sign in the expression for z is chosen to be negative when $|\theta| < \pi/2$ and positive otherwise. In all of these expressions, the values of θ are assumed to be in the range $-\pi < \theta \le \pi$.

This mapping is not a complete geometric mapping, i.e., it is not both "one-to-one" and "onto" mapping, which, in general, is not appropriate for numerical meshing.[6] However, the mapping is onto (i.e., every point in one domain maps to the other domain and *vice versa*) and is lacking the one-to-one property. This point may seem insignificant, but it is important since there are multiple points in the (x, θ) space that may map into the same point in the (x,y,z) space. In this case, it is not a problem due to the total convexity of the shapes being modeled. In fact, as a result of the total convexity, it is guaranteed that the multiple points mapping into one point will occur at exactly two locations on a convex hull, i.e., at the maximum and minimum values of x, where the hull's two-dimensional representation closes-up on itself. Knowing the exact location of these points *a priori*, a correction for the ambiguity of transforming between the two- and three-dimensional spaces at these points is made with simple logic in the computer program. Furthermore, since the mapping is onto, there is no problem of an ill-defined point in either space (such points often arise due to numerical inaccuracies) that would not transform to the other space. In fact, any such ill-defined points will transform smoothly between the two spaces (with the singular exception at the already mentioned endpoints).

In the adaptive meshing technique that has been developed, an initial mesh on the convex hull is performed using the grid spacing of 20 points in a rectangular grid in the (x, θ) space that is given by

$$x_i = \{x_{\min}, \quad (3x_{\min} + x_{\max})/4, \quad (x_{\min} + x_{\max})/2, \quad (x_{\min} + 3x_{\max})/4, \quad x_{\max}$$

$$\theta_j = \{-\pi/2, \quad 0, \quad \pi/2, \quad \pi \ ,$$

where $x_{\min}$ is the minimum value of the defined surface in the x-direction and $x_{\max}$ is the maximum value. Each of these mesh points is transformed to the full three-dimensional space, and the distance from the source to the mesh point dS_{ij} and the receiver to the mesh point dR_{ij} is calculated for each of the points in the mesh. The point whose sum of these two distances represents the minimal sum is returned as the point to use in the next iteration. For the convex shape (the only shapes under consideration), there is the

possibility that two or more of the points will share this minimal distance. However, this will occur only for neighboring points, which is another benefit of the convexity of the shapes modeled. For this reason, the iteration scheme includes nearest-neighbor points in the next iteration.

The iteration of the technique is performed in the (x, θ) space with transformations to the three-dimensional space performed with each new point added as the mesh is refined. These transformations to the physical coordinate system are the most computationally intense portion of the search algorithm, requiring one tangent evaluation and two operations of raising a real number to a noninteger power for each mesh point (in addition to numerous numerically fast operations). Keeping a regularly spaced rectangular mesh in the (x, θ) space, each iteration will require the addition of 16 new mesh points. As shown in figure 5, the new mesh includes the entire region between the chosen point of the previous iteration and all of its nearest neighbors. The three-dimensional representation of each of the new points is used to compute the source to mesh point to receiver distances, which are compared against the center point. Note that the other eight points in the refined mesh patch do not need to be evaluated since they were used at the previous step and determined to be farther than the center point. The 16 new points plus the center point that started the iteration are all compared for minimum distance and the minimum is used as the center point for the next iteration.

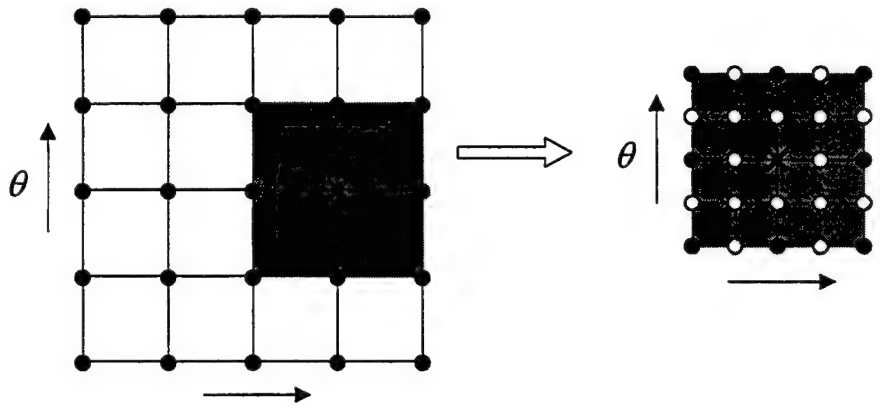

Figure 5. Diagram of the Mesh Adaptation Procedure

The iterative nature of the hull search routine allows the calculation to be fairly quick, but, more importantly, its speed of operation can be predicted without assuming anything about the convergence of any particular geometry. In practice, for a convex hull shape with length L and maximum diameter D, the nominal initial area can be assumed to be less than

$$A_0 = \pi D(L + D), \tag{11}$$

which is the area of a cylinder of those nominal dimensions with spherical endcaps. If the final location must be determined to within a wavelength and the initial area is bounded by equation (11), then, with the aid of equation (7), it can be shown that the number of iterations required is

$$n \geq \frac{\log\left(\pi D(L+D)\right) - \log\left(2\lambda^2\right)}{\log(4)}. \tag{12}$$

With this expression, the number of iterations required can be set ahead of the search by knowing only the maximum diameter, the overall convex hull length, and the wavelength (or frequency). This number of iterations for convergence is not dependent on the complexity of the shape and, therefore, provides an improvement in terms of the computational efficiency of the hull search technique. Also, the number of iterations required increases with frequency, but only as log(f), which is slower than linear for large frequencies.

As an example of the computational requirements of adaptive meshing, consider again the case of a cylinder 50 m long with a 5-m radius being ensonified by a 15-kHz pulse. Direct meshing required 157,394 mesh points, not including the endcaps. For the adaptive mesh technique, equation (12) shows that nine iterations will be required. Since 20 mesh points are generated initially and an additional 16 mesh points are generated at each iteration, this problem would require only 164 mesh points in the adaptive technique. This relatively small number of point evaluations (164 compared to more than 150,000) is what makes the adaptive meshing technique a practical approach. Direct meshing would require far too many computations.

To assess the accuracy and robustness of the adaptive mesh hull search method, comparison runs were made with exactly the same conditions as those performed for the bisection search algorithm. The timing comparison is shown in table 2 for the TestSub and Omega models.

Table 2. Performance of Adaptive Mesh Hull Search

Model	Number of Hull Searches	Total Time in Hull Searches	Average Time per Hull Search
TestSub	334	0.46 sec	1.38 msec
Omega	543	0.85 sec	1.56 msec

In contrast to the bisection hull search, the adaptive mesh hull search technique has a much more consistent average time per hull search. With this technique, the Omega model average time per hull search is only 13 percent longer than TestSub's; whereas for the bisection search, the difference was 1173 percent. However, the average time per hull search for the elliptical case of TestSub (where the bisection technique converges rapidly) is more than twice as long with the adaptive mesh hull search.

Two more models were run with both hull search techniques to confirm that the timing conclusions were consistent and not based on any peculiarities of a specific model. The two models added were called *Yellow* (a model with all elliptical hull shapes) and *XYZ* (a model with many hyperelliptical hull shapes). The results of the timing runs for all four models with both hull search techniques are summarized in table 3, which presents only the average time per hull search.

Table 3. Performance of Bisection and Adaptive Mesh Hull Searches

Model	Elliptical	Number of Searches	Bisection Average Time	Adaptive Mesh Average Time
TestSub	Yes	334	0.60 sec	1.38 msec
Yellow	Yes	297	0.68 sec	1.31 msec
Omega	No	543	7.64 sec	1.56 msec
XYZ	No	629	4.75 sec	1.86 msec

From the results shown in table 3, it is evident that the adaptive mesh search algorithm is much more consistent in runtime than the bisection search. Furthermore, the worst case of the adaptive mesh hull search (1.86 msec) is much faster than the worst case of the bisection hull search (7.64 msec). This is because the bisection hull search is much slower for nonelliptical hulls, whereas its performance for elliptical hulls is very fast. By contrast, the adaptive mesh hull search is only slightly slower for nonelliptical hulls than it is for elliptical hulls. This slowdown is attributed to the increased complexity of point transformations from two-dimensional to three-dimensional space when the exponents are not equal to two (see equation (8)).

The consistency in average hull search time that is found in the adaptive mesh search algorithm is one of its improvements over the bisection search. It should be noted that the increase in computational time using the adaptive mesh for elliptical hulls (over the bisection search) is not large enough to outweigh the performance gains found with nonelliptical hull sections.

The second performance improvement that is gained by using the adaptive mesh search algorithm is in the robustness of the computation. Running the same two cases that were shown in figures 3 and 4 (Omega model outer hull with bisection at 0.00° and 0.01° elevation angles, respectively) showed that the adaptive hull search algorithm produced a consistent result over the small change in elevation angle. The results of hull-only target strength runs for these cases are shown in figures 6 and 7. From the plots, it is

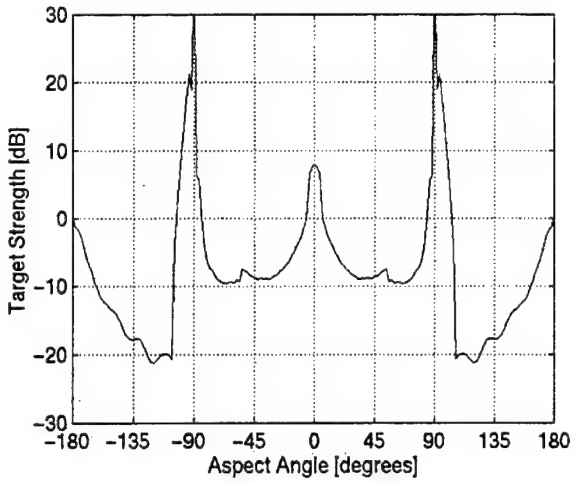

Figure 6. Target Strength for Omega Model Hull with Adaptive Mesh Hull Search at 0.00° Elevation Angle

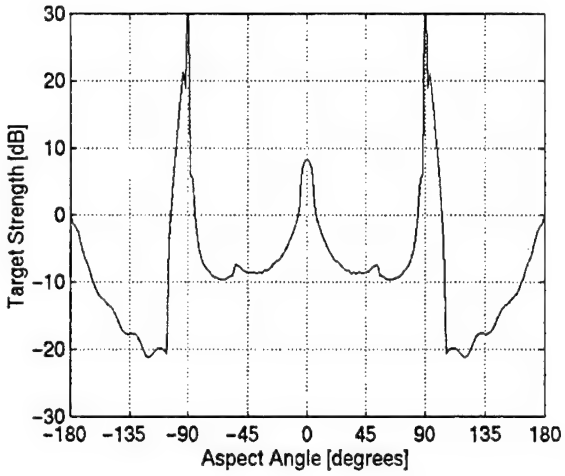

Figure 7. Target Strength for Omega Model Hull with Adaptive Mesh Hull Search at 0.01° Elevation Angle

obvious that the adaptive mesh search routine provides a robust result that does not vary drastically with extremely small angle changes.

The peculiar behavior of the target strength plot near the 0° aspect angle is a function of the squared-off nature of the bow. This type of behavior can be seen only in complex hull shapes and is not seen in simple canonical hulls. This feature is also not captured in the bisection search algorithm since the bow is at the endpoint of the two-dimensional coordinate surface and the bisection search algorithm requires many more iterations than the maximum allowed to converge there for nontrivial hull shapes. The behavior of the target strength plot in the stern section (beyond 110° aspect angle) is something that is peculiar to a complete convex hull. This effect is from the small cap-like section at the stern required to make the shape complete. If the effect is not desired, it can be removed by adding a cutout to the hull geometry definition in that region.

The accuracy of an overall acoustic scattering model such as the HFTAM will be affected by a change in the hull searching algorithm. Since the HFTAM has already been validated at the peak target strength level, the impact of any changes at the peak target strength level needs to be documented. Peak target strength calculations for the entire structure (outer hull, sail, internals, etc.) were computed for the HFTAM with the bisection hull search routine and with the adaptive mesh hull search routine for all four targets studied. The results of the peak target strength comparisons can be seen in figures 8 through 15. All of these results were plotted in a polar format with a radial direction displaying level in decibels and the rotational direction representing aspect angle (off-bow) in degrees.

As can be seen in figures 8 through 11, the two models that had all elliptical hull cross sections (TestSub and Yellow) produced similar peak target strength results for the two different hull search techniques. This is expected because the bisection algorithm has no problems with purely elliptical hull geometries. The adaptive mesh hull search technique always converges to a point that is near the true specular point, so it is believed to be a more accurate technique. However, the existing HFTAM with the bisection search algorithm has also been validated against acoustic data and, therefore, must be

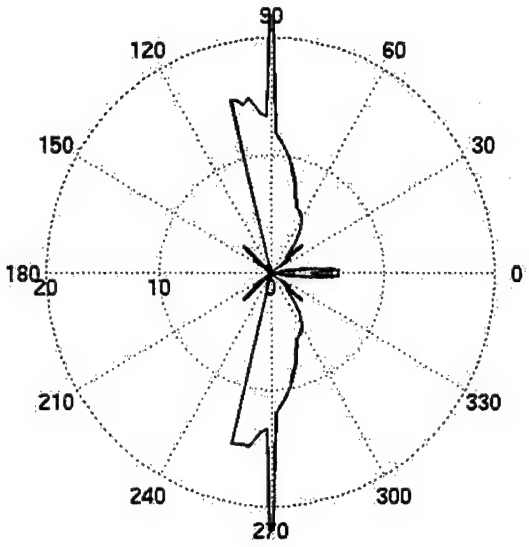

Figure 8. Peak Target Strength of TestSub Model Using Bisection Hull Search Technique

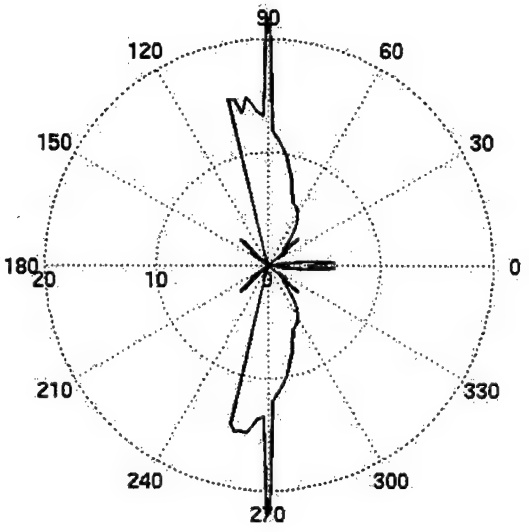

Figure 9. Peak Target Strength of TestSub Model Using Adaptive Mesh Hull Search Technique

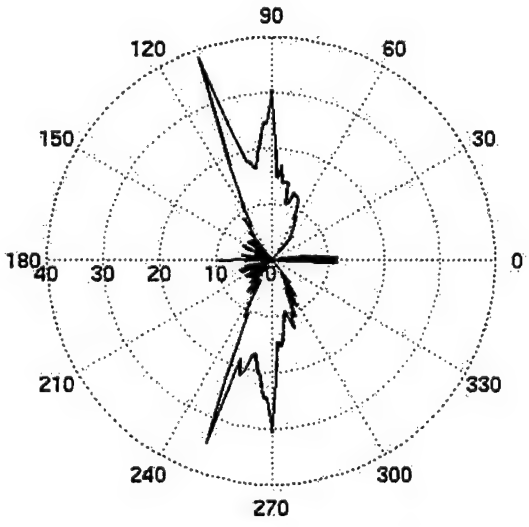

Figure 10. Peak Target Strength of Yellow Model Using Bisection Hull Search Technique

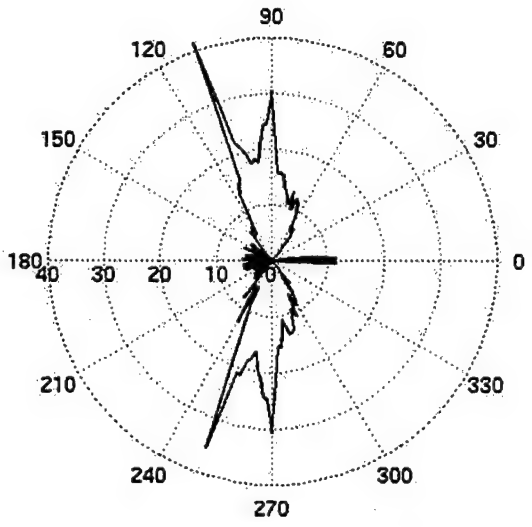

Figure 11. Peak Target Strength of Yellow Model Using Adaptive Mesh Hull Search Technique

assumed to be accurate for the shapes that have been used for the model validation. The existing validations were performed with target shapes that are purely elliptical, so a comparison of figures 8 and 9, as well as figures 10 and 11, shows that existing validation data are still applicable to the adaptive mesh hull search algorithm. Thus, the adaptive mesh hull search algorithm is at least as valid as the bisection algorithm, when compared with previous validations.

For nonelliptical hull geometries, such as those found in the Omega and XYZ models, the adaptive mesh algorithm computes a specular point for all cases. However, there are times when the bisection algorithm fails to converge in the Omega and XYZ models and, therefore, returns no target strength. Thus, the total net peak target strength for all highlights may, for certain aspects, be lower in the bisection algorithm than in the adaptive mesh algorithm. The plots in figures 12 through 15 confirm this. In figures 12 and 13, for the Omega model, the difference in peak target strength is very small between the two techniques. This is due to the hull not being the dominant highlight over most aspect angles (the exception being near bow and beam). In figures 14 and 15, for the XYZ model, there is a considerable difference in peak target strength between the two techniques, which can be attributed to the dominance of hull highlights for this particular target. Also, the behavior near the stern (180°) that is exhibited in figure 15 is caused by computing the reflection near the tail section of the outer hull, a region where the bisection algorithm fails to converge (see figure 14).

The discrepancies found between the two techniques in figures 14 and 15 should not affect any existing validations of the HFTAM since these are based on the XYZ model. The XYZ model has not yet been validated at the peak target strength level against acoustic data; therefore, it is not imperative to make the two techniques agree at

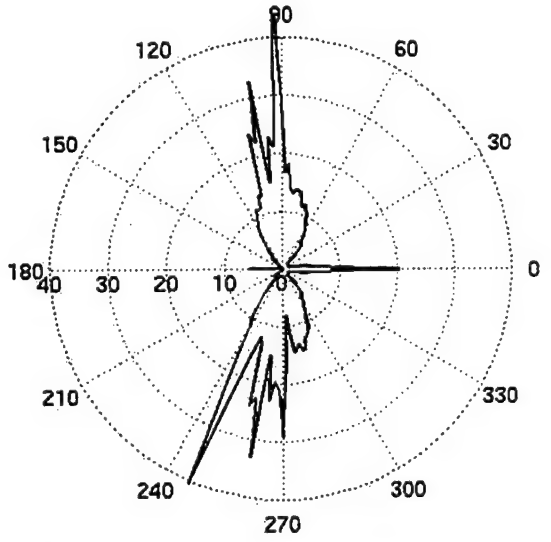

Figure 12. Peak Target Strength of Omega Model Using Bisection Hull Search Technique

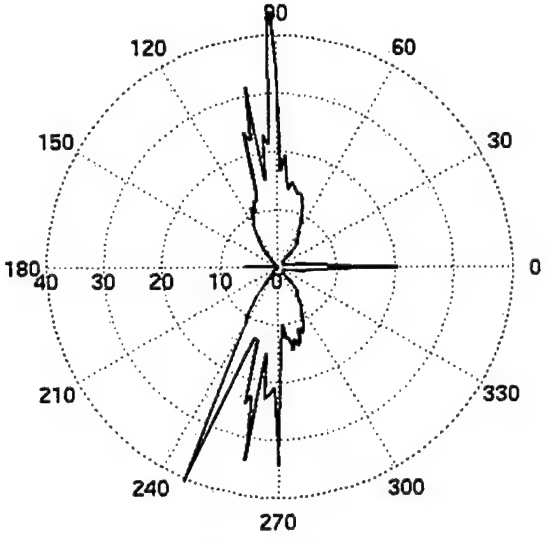

Figure 13. Peak Target Strength of Omega Model Using Adaptive Mesh Hull Search Technique

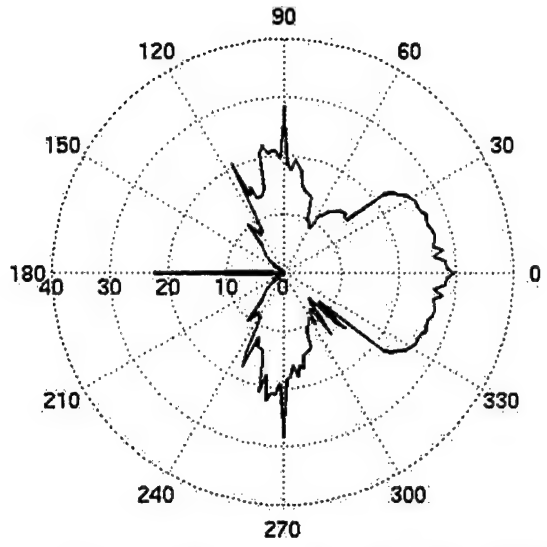

Figure 14. Peak Target Strength of XYZ Model Using Bisection Hull Search Technique

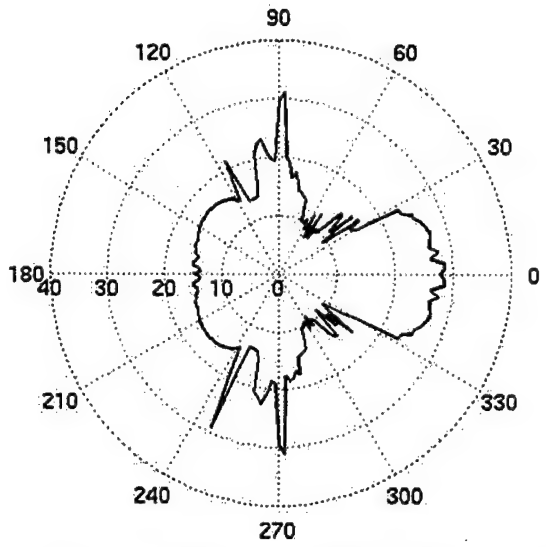

Figure 15. Peak Target Strength of XYZ Model Using Adaptive Mesh Hull Search Technique

the peak target strength level. It is anticipated that the results with the adaptive mesh hull search technique (figure 15) will be more accurate than the results for the bisection hull search technique (figure 14).

The peak target strength validation metric is not the only validation metric available, but it has been presented here to show the validity of the new hull search technique. This analysis provides a connection with the existing validation databases. The new technique should also be validated by specific users with regard to their needs, such as highlight location, time echo characteristics, etc. The comparisons shown in this report are illustrative only and do not represent a complete validation.

SOFTWARE CHANGES FOR IMPLEMENTATION

To implement the adaptive mesh specular point search algorithm, a number of changes needed to be made to the HFTAM software. These changes have all been made and tested on HFTAM (Version 1.5) to produce the results shown in this report. It is expected that the adaptive mesh hull search routine will become the default hull search technique in the next version of the HFTAM (Version 1.6).

The HFTAM software changes were all written in ANSI C and conform to the programming style used in the other HFTAM routines. (All of the software changes that are documented in this report are maintained by the author of this report.) The programs were tested by creating a new model routine bisim2.c that was very similar to the existing bisim.c with the exception of calling up the new hull search in place of the old hull search. Table 4 lists the programs that were created for the new hull search routine.

Table 4. Additional HFTAM Software Routines Written for the Adaptive Mesh Implementation

Name	Description
HullSeach2.c	Main program for adaptive meshing used in the same way as HullSearch.c.
PatchEval.c	Finds the point with minimal distance for a given level of mesh refinement.
Xtheta2XYZ.c	Computes the three-dimensional coordinates of a point from its two-dimensional representation.
ddelli2.c	Computes parameter values only (without derivatives) for a fairing ellipse section.
ddpoly2.c	Computes parameter values only (without derivatives) for a polynomial section.
GetFitParameters2.c	Finds (a,b,d,p) for a given x value.

Table 5 lists the routines that required modification to accommodate the new hull search routine.

Table 5. HFTAM Software Routines That Are Changed for the Adaptive Mesh Implementation

Name	Description
Echoes.c, bisim.c	Recompiled to be linked with new routines.
m_protos.h	Modified to include the input/output and call-list structure of each of the new routines shown in table 4.
Hull.c	Changed to call HullSearch2 instead of calling HullSearch.

When the new hull search routines are in place, some old search routines will become obsolete. Routines that were specific to the bisection hull search routine and that will no longer be required in an implementation of HFTAM are listed in table 6.

Table 6. Software Routines That Become Superseded in the Adaptive Mesh Implementation

Name	Description
GetCandidatePoint.c	Finds (y,z) coordinates for a sample x value.
HullSearch.c	Replaced by HullSearch2.c.
YZEqn.c	Performs the bisection search on a fixed x slice for the analytical distance minimization problem.

CONCLUSIONS

A new adaptive mesh technique for searching for a bistatic specular point on a convex hull surface has been developed for the NUWC HFTAM. The existing method of bistatic specular point searching in HFTAM uses a bisection search with numerical evaluations of a transcendental equation at each bisection step. This method has been shown to lack computational robustness.

The adaptive mesh technique has been implemented with the HFTAM and has been shown to be more robust than the bisection technique. Furthermore, the speed of computation for the new technique is independent of the complexity of the geometric shape being modeled. The bisection technique, in contrast, was shown to perform very slowly when nonelliptical hull sections were used. For many of the complex shapes that are defined to a high level of geometric fidelity, the nonelliptical representations are necessary and, thus, there is a need for the new hull search technique.

The adaptive mesh hull search technique is expected to become the hull search used in a future release of the HFTAM. It has been shown to improve robustness of the current technique and to improve the speed of the search process for complex geometries. The impact of the new search routine on the HFTAM has been shown in terms of overall peak target strength as well as some individual hull target strength comparisons. The model shows results that are at least as accurate as the existing technique. These improvements in the hull search routine will improve the speed and fidelity of the HFTAM for future simulation studies.

REFERENCES

1. D. H. Von Seggern, *CRC Standard Curves and Surfaces*, CRC Press, Boca Raton, FL, 1993.

2. G. A. Lengua, "Modeler's Reference Guide to the NUWC High-Frequency Target Acoustic Model," NUWC-NPT Technical Report 10,532, Naval Undersea Warfare Center Division, Newport, RI, February 1996 (UNCLASSIFIED).

3. G. A. Lengua, "Bistatic Specular Scattering from Hyper-Elliptical Cross-Sections," NUWC-NPT Internal Memorandum 38423/48, Naval Undersea Warfare Center Division, Newport, RI, August 1993 (UNCLASSIFIED).

4. R. L. Burden and J. D. Faires, *Numerical Analysis*, PWS-Kent, Boston, MA, 1988.

5. J. B. Keller, "Geometric Theory of Diffraction," *Journal of the Optical Society of America*, vol. 52, February 1962, pp. 116-130.

6. F. S. Acton, *Numerical Methods that Work*, Mathematical Association of America, Washington, DC, 1990.

APPENDIX A
HYPERELLIPSE PARAMETERS FOR SAMPLE CONVEX HULL OF FIGURE 1

This appendix contains the equations defining the hyperellipse parameters for the surface shown in figure 1. The convex surface of this shape is defined on the region $-55.1688 \le x \le 55.1688.$. The equations defining the surface are the generalized hyperellipse equations in the form of equation (2), and the parameter a is given by the following:

$$a = \begin{cases} 8.7558 + 0.47683x + 0.021646x^2 \\ \quad +\left(4.2292 \times 10^{-4}\right)x^3 + \left(2.4481 \times 10^{-6}\right)x^4 & \text{for } -55.1688 \le x < -31.5, \\ -3.93041 + \sqrt{-0.02805x^2 - 0.66686x + 76.31107} & \text{for } -31.5 \le x < -11.8872, \\ 5.0292 & \text{for } -11.8872 \le x < 36.8808, \\ 2.9442 + \sqrt{-0.027338x^2 + 2.01651x - 32.8381} & \text{for } 36.8808 \le x < 44.5069, \\ -22.027765 + 1.32234x - 0.016266x^2 & \text{for } 44.5069 \le x < 49.9933, \\ \sqrt{-0.35705x^2 + 1.48663x + 26.65693} & \text{for } 49.9933 \le x \le 55.1688. \end{cases}$$

For this particular shape, $b \equiv a$ for all x. Furthermore, for this shape, the parameters d and p satisfy the following:

$$d = 0, \quad \text{for } -55.1688 \le x \le 55.1688$$
$$p = 2, \quad \text{for } -55.1688 \le x \le 55.1688.$$

Therefore, all of the cross sections of the shape are circles (since $a = b$ and $p = 2$) that are centered about the x-axis (since $d = 0$).

APPENDIX B
HYPERELLIPSE PARAMETERS FOR SAMPLE CONVEX HULL OF FIGURE 2

This appendix contains the equations defining the hyperellipse parameters for the surface shown in figure 2. The convex surface of this shape is defined on the region $-9.89 \le x \le 11.0$. The equations defining the surface are the generalized hyperellipse equations in the form of equation (2). The parameter d for this shape is varying, and therefore, the centerline of the cross sections is nonconstant throughout the x-range. In this case, the d parameter takes on the value

$$d = \begin{cases} -3.01835 + \left(3.7245 \times 10^{-3}\right)x - \left(7.5864 \times 10^{-4}\right)x^2 & \text{for } -9.89 \le x < -2.2307, \\ -2.8479 + 0.53107x + 0.62639x^2 & \\ \quad + 0.35628x^3 + 0.09839x^4 + 0.01064x^5 & \text{for } -2.2307 \le x < -1.3135, \\ -3.026132 & \text{for } -1.3135 \le x \le 11.0. \end{cases}$$

Using this variable d parameter, the cross sections are defined in two components. The first components are labeled with a subscript 1 (i.e., a_1, b_1 and p_1) and are valid for points where $z \ge d(x)$. The second components are labeled with a subscript 2 (i.e., a_2, b_2, and p_2) and are valid for points where $z \le d(x)$. To have a smooth (up to C^2 smoothness is guaranteed) junction between these two parts of each cross section, the b parameters are forced to be the same, $b_1 = b_2$.

For the convex hull shape in figure 2, the cutout region will be defined to contain the entire region of the subscript-1 parameters (where $z \ge d(x)$), as well as a portion of the subscript-2 parameters. Thus, the values of the subscript-1 parameters are somewhat arbitrary. For convenience, the values of the a parameters can be forced to be the same by making a_1 the same as a_2. Also, for further simplicity, the value of p_1 is assumed to be constant at $p_1 = 2$ throughout the entire x-domain. The parameter a_1 (which for this shape is the same as a_2) is given by the following expression:

$$a_1 = \begin{cases} -72.5113 - 15.8452x - 0.86036x^2 & \text{for } -9.89 \le x < -9.75, \\ -333247.6 - 176750.3x - 37483.41x^2 - 3972.92x^3 \\ \quad -210.459x^4 - 4.4575x^5 & \text{for } -9.75 \le x < -9.175, \\ 1.38480 - 0.054875x - 0.017189x^2 & \text{for } -9.175 \le x < -2.23074, \\ 1.96119 + 1.6046x + 1.9074x^2 + 1.10659x^3 \\ \quad +0.31345x^4 + 0.034934x^5 & \text{for } -2.23074 \le x < -1.3135, \\ 1.43304 & \text{for } -1.3135 \le x < 9.4, \\ 50240.1 - 24949.4x + 4948.6x^2 - 490.038x^3 \\ \quad +24.2284x^4 - 0.47851x^5 & \text{for } 9.4 \le x < 10.7, \\ -6.8818 + 2.3587x - 0.15923x^2 & \text{for } 10.7 \le x \le 11.0. \end{cases}$$

The last parameter that remains to be defined for this convex hull shape is p_1. The parameter p_1 is defined by the following expression:

$$p_1 = \begin{cases} 22.1338 + 0.94835x - 0.31932x^2 \\ \quad -0.040468x^3 - (1.93717 \times 10^{-3})x^4 & \text{for } -9.89 \le x < -2.2307, \\ 23.3906 + 11.8417x + 19.1948x^2 + 14.2006x^3 \\ \quad +4.67320x^4 + 0.57471x^5 & \text{for } -2.2307 \le x < -0.8135, \\ 20.657 & \text{for } -0.8135 \le x < 5.3736, \\ 40279.12 - 33263.6x + 10932.12x^2 - 1786.003x^3 \\ \quad +145.033x^4 - 4.68386x^5 & \text{for } 5.3736 \le x < 6.3718, \\ 784.119 - 340.423x + 56.1141x^2 \\ \quad -4.11922x^3 + 0.11312x^4 & \text{for } 6.3718 \le x \le 11.0. \end{cases}$$

The preceding expressions define a complete convex hull shape. To obtain the shape found in red in figure 2, the cutout region needs to be defined. The cutout region for this convex hull shape is defined by all values of z that are greater than the following:

$$z > \begin{cases} -3.02819 & \text{for } 9.73777 \le x < 21.63, \\ -5.36825 + 0.10819x & \text{for } 21.63 \le x < 27.35, \\ 0.0011 + 00154x + 0.1448x^2 \\ \quad -0.0115x^3 + 0.00022x^4 & \text{for } 27.35 \le x < 29.75, \\ -6.7875 + 0.02169x & \text{for } 29.75 \le x < 30.25, \\ -0.2265 & \text{for } 30.25 \le x \le 30.95. \end{cases}$$

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Program Executive Officer, Undersea Warfare PMS-404: M. Jacobs, CDR D. Shafer	2
Defense Technical Information Center	12